Hardware prototype with component specification and usage description

D3.2 Hardware prototype with component specification and usage description

Editors: Tre Azam (MP)
Soyeb Aswat (MP)

Wearable Experience for Knowledge Intensive Training
Project No 687669

Revision History

Version	Date	Contributor(s)	Modification
0.1	09-11-16	Tre Azam (MP)	Initial Outline and Review of D1.3, D1.4, D5.1 and related deliverables
0.2	10-11-16	Tre Azam (MP)	Draft Intro, Executive Summary, Spec table, Usage descriptions
0.3	10-11-16	Soyeb Aswat (MP)	API, SPU Diagram, Other Contributions
0.4	10-11-16	Tre Azam (MP)	Pricing Matrix, Improvements to other areas
0.5	13-11-16	Tre Azam (MP)	Review and Edit, Added Prototype Design
0.6	21-11-16	Tre Azam (MP)	Update and additions as per feedback
0.7	24-11-16	Soyeb Aswat (MP)	Comparison of push and pull interfaces. Small fixes to API definition.
0.8	28-11-16	Tre Azam (MP)	Update Prototype image and add Usage Description
0.9	28-11-16	Mikhail Fominykh (EP)	Applying WEKIT style
1.0	28-11-16	Tre Azam (MP)	Final fine tuning

Hardware prototype with component specification and usage description

- First version

WP 3 | D3.2

Editors:

Tre Azam (EP)

Authors:

Soyeb Aswat (MP)
Roland Klemke (OUNL)
Puneet Sharma (UiT)
Fridolin Wild (OBU)

Reviewers:

Mikhail Fominykh (EP)
Fridolin Wild, OBU
Ralf Klamma, RWTH

Deliverable number	D3.2
Dissemination level	Public
Version	1.0
Status	Final
Date	28.11.2016
Due date	30.11.2016

Table of Contents

Executive summary

Following on from D3.1 and the final selection of sensors, in this D3.2 report we present the first version of the experience capturing hardware prototype design and API architecture taking into account the current limitations of the Hololens not being available until early next month in time for integration into the current proposed framework and design.

This deliverable involved acquiring, testing and integrating the various off-the-shelf sensors and developing a hardware/software design to connect the various devices and sensors into a single platform. This also involved solving problems around how we would manage the additional computing power, storage and wireless streaming capabilities required for this project.

In this first version of the deliverable we propose an initial design for the hardware and architecture and have focussed initially on providing a bare-bones prototype of the Sensor Processing Unit (SPU) which involved combining a micro-pc with various sensors to test the flow of data, capabilities of the system and possible connection methods to the device for the various sensors.

The first hardware prototype does not yet include the wearable element, however, our selection of the sensors and micro-pc takes into consideration the desire to integrate the hardware using 3D printing or by creating add-ons for the Hololens to avoid a chunky or multi-device hardware solution; D5.2 (M15) will be providing the final design of the fashion, wearability, and comfort of the device taking into consideration the previously completed desk based research (D5.1, M15). We have provided a mock-up in the Prototype and Usage Descriptions section below of what the final prototype is expected to look like.

The separation of the Sensor Processing Unit (SPU) allows for the development of the hardware and software to continue in the absence of the Hololens, but also ensures the platform can be used standalone or is easily adapted to other Smart Glasses. It does not add any further strain on the in-built computers by keeping the sensor processing independent.

As described in D3.2 although there are other AR alternatives already available in the market, they all fall short in terms of the required spec and functionality. A decision was then made to work with the Hololens as it would be the best in class and although not available has been tested by key partners and will be available in time for the next deliverable.

Within this deliverable we also provide an initial mock-up of the hardware design and also a sample of the API and links to device APIs.

1. Introduction

As outlined in the WEKIT project description, the objective of this deliverable (D3.2) is to create a wearable hardware platform for capturing the user experience based on sensor specifications outlined in D3.1, and provide more detailed specifications on the hardware selected and how they integrate together to form the final wearable device.

This deliverable also involves reviewing previous deliverables D1.3, D1.4, and D5.1 which have been used to provide guidance based on case scenarios, wearability and compatibility taking into account other concerns such as possible noise and interference to sensors.

Additional Technical requirements were identified in D1.1/D1.2 (Section 6.3. Technology Needs, p.46), esp. insight T5

In section 2 of D3.2 we start by providing a graphical breakdown of the proposed structure for the hardware/API which will feed into deliverable D3.3 and more specifically how the Sensor Processing Unit (SPU) will connect into the Hololens hardware to complete the Hardware requirements.

In this section we also include details of the micro-pc selected to provide the storage, power and sensor acquisition module with the proposed method for communications between devices.

In section 3, we provide a compiled list of the selected hardware and possible alternatives pending further testing, which will go into the first hardware prototype deliverable when the Hololens is available.

In section 4, we provide a simple list of usage guidelines and restrictions, from placement of sensors to possible battery and processing time limitations, amongst other usage descriptions.

This deliverable document, along with the initial hardware prototype, should provide partners from WP2, WP3, and WP4 the information on sensors, data outputs, connection methods and the drivers required to continue the development of the hardware/software API and re-enactment data visualisations.

The final output and user testing will also require input and collaboration from WP5 for human factors and wearability.

2. Proposed Architecture

In this section with Figure 1, we present the proposed hardware architecture as cohesion of different high-level components. In Figure 2 we elaborate on the Sensor Processing Unit and highlight the various connection methods for the hardware API. The associated software architecture for this platform has been proposed in the deliverable D2.1 (Fig. 1).

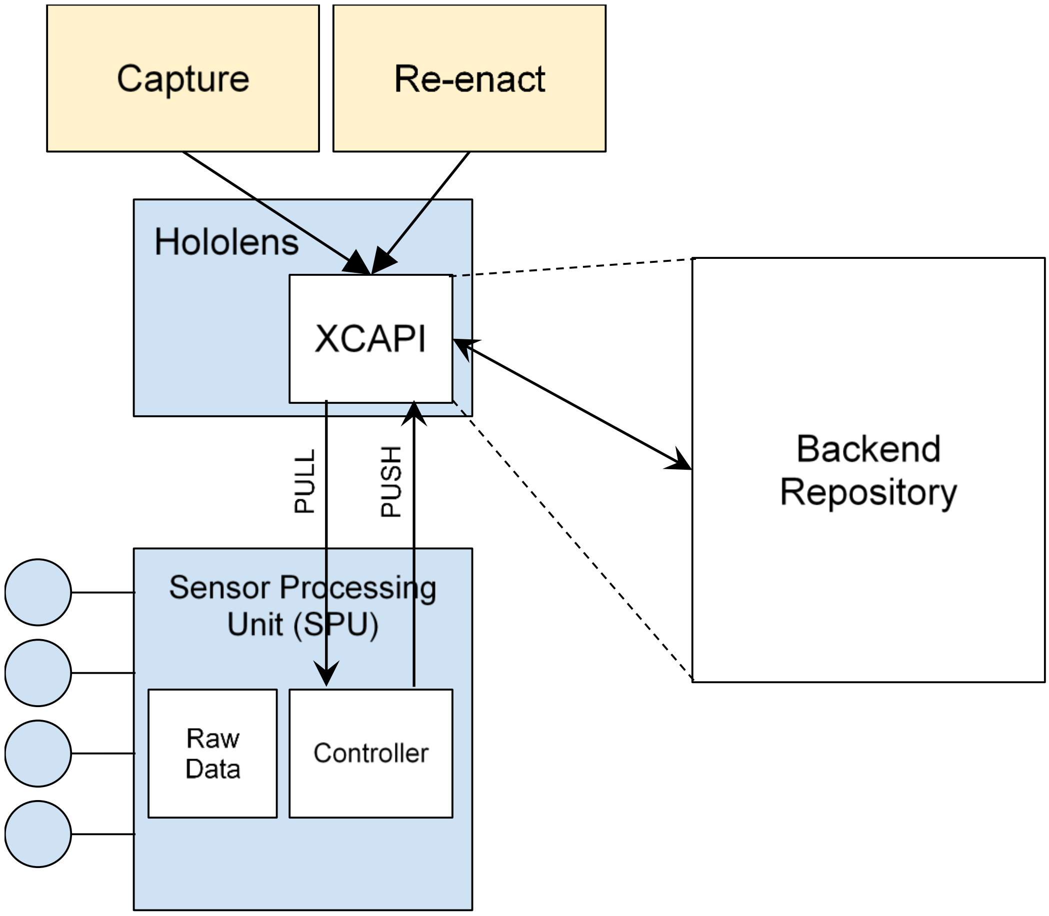

Figure 1. Proposed design architecture for D3.2 & D3.3. The software architecture is further elaborated on in D3.3. The backend repository will be used to store the data from Sensor Processing Unit, and Hololens.

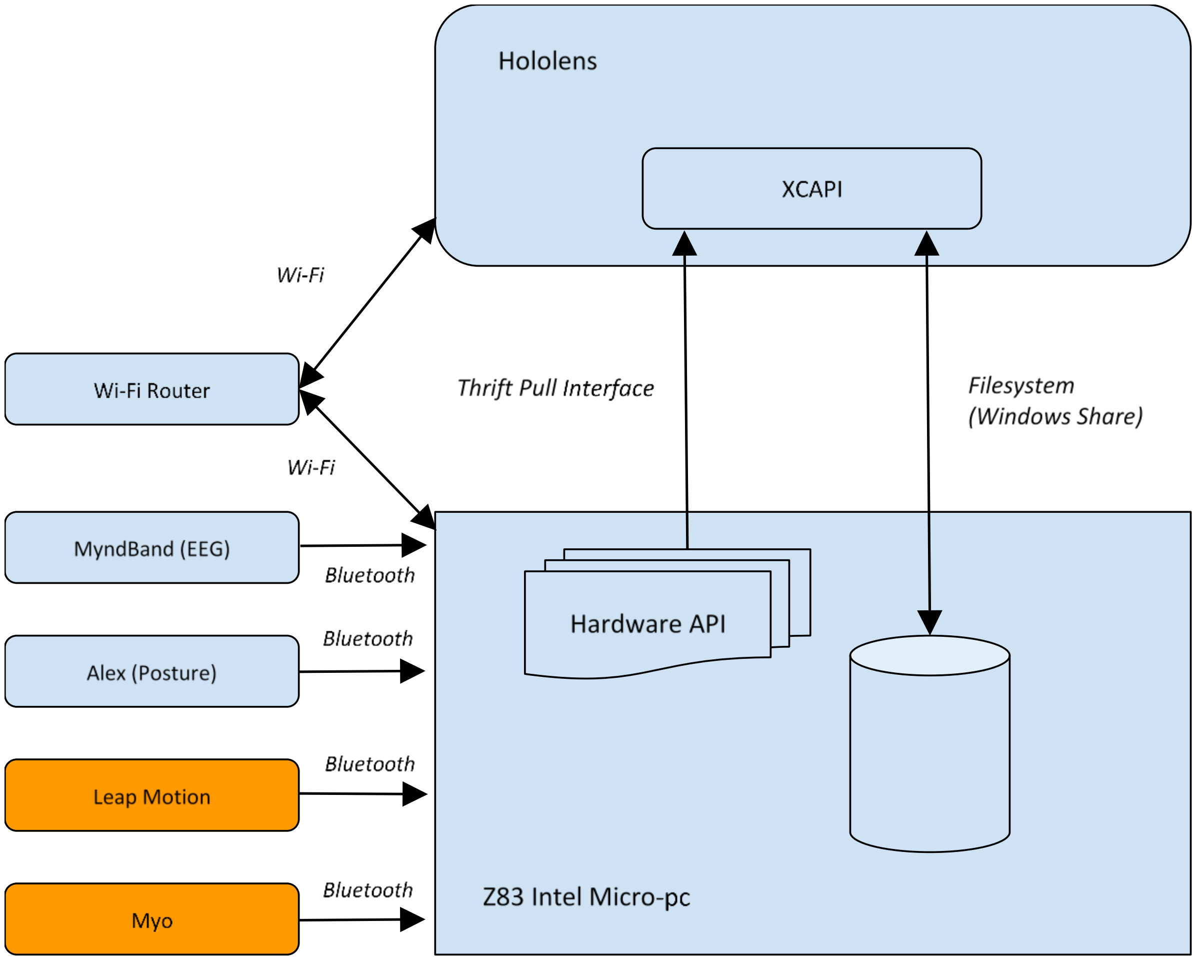

Figure 2. This figure shows the different communication modes (Wi-Fi, Bluetooth) employed of the different sensor components. The choice of the different communication modes is governed by data bandwidth requirements, technical specifications of the sensors and the design requirements. Intel Micro-pc will serve as processing unit to control sensors, and read/write sensor data. The hardware API is a Thrift-based front-end to the underlying drivers.

3. Component Specifications

In D3.1, a final selection of sensors was identified and recommendations were made. From the initial list we have highlighted below the devices we have incorporated into the first hardware prototype. The sensors in orange are under consideration and the ones in red for phase 2, the sensors in green are being incorporated in the first phase.

Table 1. Selected components, specifications and links to current API - Ref D3.3 for software requirements of sensor components

Sensors and Supporting Devices	Technical Specifications	API/Drivers
Lenovo Ideacentre Stick 300 *First Choice due to size and weight	Intel Atom Z3735F 1.33 GHz, 2 GB RAM, 32 GB eMMC, WLAN, Bluetooth, Integrated Graphics, Windows 10, very compact and lightweight with expandable storage	NA or integrated
Z83 Intel Micro PC *Bulky but more powerful alternative	Z83 Windows 10 4K Intel Atom X5-Z8300 Quad-core 4K 1000M Ethernet 64 Bit 2.4G + 5.8G Wi-Fi BT4.0 Connectivity LAN	NA or integrated
TP-LINK TL-MR3020	2.4 GHz, 150 Mbps, USB 2.0, Travel Router to provide access point for Hololens and SPU	http://www.tp-link.com/en/download/TL-MR3020.html
TP-LINK M-7350 *Alternative with built in battery but heavier	Wi-Fi router with built-in battery. Router to provide access point for Hololens and SPU	http://www.tp-link.com/en/download/M7350.html
MyndBand and Neurosky chipset	Raw-Brainwaves, EEG power spectrums (Alpha, Beta, etc.), Attention, Meditation, and other future metrics, detect poor contact, and whether the device is off the head.	Available direct from MP
ALEX Posture	Bluetooth neck posture tracker with long life battery and easy fit integration and haptic feedback option	Requested
Additional Battery to power Mini PC	Currently reviewing power requirements	NA
Leap Motion	Range: 1m, Frame rate: 200 fps, precision: 0.7 mm.	Available but made obsolete by the Hololens
Myo	Sampling data rate for electromyography is 200 Hz, sampling date rate for Inertial Sensor is 50 Hz.	https://developer.thalmic.com/docs/api_reference/platform/the-sdk.html
Augmented Reality Glasses (Microsoft Hololens)	See-through holographic lenses (waveguides, 2 HD 16:9 light engines, Automatic pupillary distance calibration. Holographic resolution: 2.3M total light points, holographic density: >2.5k radiant (light points per radian), 4 environment understanding cameras, inertial measuring unit, depth camera.	

Augmented Reality Glasses (ODG R-7)* Alternative to Hololens	Dual 720p Stereoscopic See-thru displays at up to 80fps, 80% See-through transmission, Magnetic Removable Photochromic Shields.	http://www.osterhoutgroup.com/developers
Built-in microphone (Microsoft Hololens)	4 microphones	
Gaze (Microsoft Hololens)	Gaze cursor	
Intel RealSense	Range: front-facing: 20-180 cm, rear-facing 50-500 cm. Depth camera with 640x480 resolution (30 fps)	
ECG Heart Rate	Heartmath or Generic for Raw and Processed data	
Eye Tracking	Under Review	

4. Quality Requirements

A key technical requirement will be managing interference, limitations and environmental factors that may affect the overall quality of the data, signal and or the user experience. In this section we have identified some areas of concern for further investigation during partner trials.

Limitations and Environment:

- Temperature
- Rain, snow, and wind
- Electrical noise or electromagnetic interference
- Ventilation
- Light and sound levels
- Local Wi-Fi network at site

The data transfer has the following requirements:

1. Serialisation of data
2. Multiple OS compatibility
3. Language compatibility with C++ and C#
4. Transparent transfer locally or remotely

Two libraries were evaluated; Apache Thrift (https://thrift.apache.org/), and Google Protocol Buffers (Protobuf) (https://developers.google.com/protocol-buffers/). Both provided the first three requirements, Thrift has the fourth built-in, whilst Protobuf requires an extension. Ultimately Apache Thrift was chosen as it has RPC connectivity built-in.

Thrift only provides a pull interface. A pull interface relies on the user of the interface to request information. The benefit of this is the interface does not need to retain any state about its users, for example what events a user wishes to be notified about, thereby simplifying the implementation. In our use case, the Thrift interface would have to be polled at regular intervals to ask if any new information from the device is available. A push interface in contrast can notify the user of events directly, for example when new data is available.

In the first prototype, data will only be available for pulling. In the future it may be possible to add facilities to simplify creating a push interface on the client end.

Thrift has its own interface definition language (IDL). Interfaces will be described using this language. The interface definition can be seen in Annex 1.

5. Prototype and Usage Description

5.1. WEKIT prototype mock-up

The image below shows a mock-up of the WEKIT prototype (Fig. 3). Components 2-8 combined are what we refer to as the Sensor Processing Unit (SPU).

Figure 3. Mock-up of proposed design with selected sensor components, fitting and placement

5.2. Usage Description and Wearability notes

Wearability and human factors are vital for the successful uptake of projects such as WEKIT; in the section below we breakdown some of the design thoughts and required behaviours to ensure the platform and hardware are suitable, comfortable, and providing the best possible quality of data for the user.

- Micro-pc will be housed in a mould to clip onto the side of the Hololens, an additional battery will act as a counterweight on the other side for additional power, this is to ensure the device does not become too heavy or slide to one side.
- The EEG sensors must be accurately placed so it will have Velcro place-markers within the Hololens forehead strap to ensure the correct placement of brainwave sensor in relation to the positioning of the Hololens itself.
- The Alex Posture band will also attach directly to the Hololens via a mould, eliminating the requirement to attach the sensor to the ears causing discomfort and reducing mobility.
- The Leap or Myo will be the only device not directly integrated into the hardware, and will connect via Bluetooth and be paired in advance to the allocated micro-pc and should be numbered for convenience.
- Due to the number of devices, a charging dock or similar may be required to ensure the paired and connected devices do not get mixed up and parts of the system do not stop working due to low battery issues. We must also consider a central battery level visualisation or low battery notification from the various sensors.

The Hololens and SPU will also require separate charging facilities however with the SPU it is possible to have a removable and replaceable external battery for smoother transition between charged devices.

6. Discussion

("Insight I3: The appropriate technology is more important than price.", deliverable D1.1/D1.2)

Our focus is to select the appropriate technology (best in class). Prices then can be negotiated down with volume.

Additional points of discussion relating to the hardware design and architecture are noted below:

- Do we add a wireless access point to the system to ensure communications between components without a network available?
- Where is calibration required and at what stage will this be done?
- What is the desired connection method between the Hololens and SPU?

- With current use cases and trial partners what are the wearability limitations on site?
- What is the expected battery life of the completed product?

Trials will show whether some of the body-worn (inside-out) sensors fulfil the needs expressed in user requirements. It should, however, be noted that for e.g. posture there are alternative solutions, such as fixed-installed room equipment like the Microsoft Kinect.

Table 2. Pricing matrix of selected sensors for first prototype: Prices correct on date 10/11/2016

Sensors	Availability or Alternatives	Current Retail Price
Hololens	Available in the UK from Dec 2016	£2719 Dev
Micro-PC	Lenovo Ideastick or alternative	£99-£129 retail
Additional Battery	A number of options being tested	£99 retail for 40k mA
Alex Posture	Available	£79-£99 retail
MyndBand EEG Headset	Available	£179 retail
Myo Movement Tracker	Available	£158 at current FX rate
Wi-Fi Router	TP-LINK M-7350 Available	£69.99 -£89.99
ECG	TBC	£29-£99 options

7. Conclusions

The goal of D3.2 is to finalise the design, functionality and core architecture of the first physical prototype so that we are able to get practical experience of the sensors and their suitability for the task with the selected real-world trial scenarios.

Further testing is required to test and validate the connectivity and compatibility of the various Bluetooth devices and the computing power required to attain the desired objectives in storage, power, and high fidelity sensor data acquisition. Another key testing point is the local power consumption of the SPU; this will also dictate the minimum requirements for the additional power bank.

The arrival of the Hololens will also add the additional complexity of managing and synchronising the various data streams into a single recording timeline.

- Current planned roadmap: iterations of this deliverable (final version in M27: D3.5)
- Note: the evaluation trials are due in M18, so the first version should be ready for implementation by this time and will then be adapted, improved and refined between M18-M27.

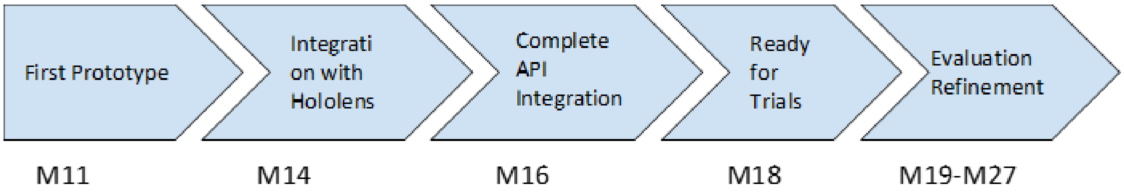

Figure 4. Timeline plan for this deliverable

The final deliverables will feed into the next deliverables D2.4, D2.5, D3.3, D4.3, D5.1, D5.2 and WP6 when the prototype is ready for case trials.

The current prototype does not include the Hololens at all involved partner sites and is therefore not yet ready for trials. Once all key partners receive the Hololens and can integrate the designs, we will be ready for trials.

Feedback required from evaluation trials:

- Wearability, comfort, practicality
- How long can you wear it?
- Visual fatigue
- Battery life requirements
- Noise or interference issues
- Usefulness for required task

Annex 1. Thrift API description

Draft description of the Thrift API.

```
// updateTime = time this update was created
// signalLevel = quality of EEG signal (0-100)
// attention, meditation = focus and relaxation level (0-100)
// lowAlpha, highAlpha etc. = brainwave channels

struct EEGOneSecondValues {
1: i64 updateTime,
2: i8 signalLevel,
3: i8 attention,
4: i8 meditation,
5: i32 lowAlpha,
6: i32 highAlpha,
7: i32 lowBeta,
8: i32 highBeta,
9: i32 lowDelta,
10: i32 highDelta,
11: i32 lowGamma,
12: i32 midGamma,

}

// rawValues = list of raw EEG signal values (+/-32768). struct EEGRawForOneSecond {
1: i64 updateTime,
2: list<i16> rawValues,

}

// Values are updated once per second and available here.
service EEG extends shared.SharedService {
EEGOneSecondValues oneSecondValues(),
EEGRawForOneSecond rawForOneSecond(),

}

// updateTime = time of last update in milliseconds

struct BodyPostureValues {
1: i64 updateTime,
// real posture values

}
```

```
// The last posture value with the time it was collected.
service BodyPosture extends shared.SharedService {
PostureValues lastPosture(),

}

struct SingleHandPosture {
// hand posture values

}

// updateTime = time of the last posture snapshot in ms
struct HandPostureValues {
1: i64 updateTime,
2: SingleHandPosture left,
3: SignleHandPosture right,

}

service HandPosture extends shared.SharedService {

HandPostureValues oneSecondValues(),

}

struct Coords {
1: float x,
2: float y,
3: float z,

}

// updateTime = time when in this position in ms
struct PositionalValues {
1: i64 updateTime,
2: Coords location,
3: Coords orientation,
4: Coords gaze,

}

service Position extends shared.SharedService {
Positional position()

}

// Get time on remote machine.  Used for synchronisation.
struct CurrentTimeValues {
```

```
1: i32 currentTimeS,
2: i64 currentTimeMS,

}

// Get time on remote machine.  Used for synchronisation.
service CurrentTime extends shared.SharedService {     CurrentTimeValues currentTime()

}
```

Wearable Experience for Knowledge Intensive Training
Project No 687669

WEKIT project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687669. http://wekit.eu/